Amendment to the Specification

(On page 6, lines 11-12) Please replace

Design Principle 1: Make it very quick and easy to select which portion of the strategy is displayed on screen

with

Design Principle 1: Make it very quick and easy to select which portion of the strategy is displayed on screen.

(On page 9, lines 15-25) Please amend as follows:

The invention herein treats the focus node specially. ~~It's~~ Its sibling segments are displayed without any of their descendant segments. The invention herein also uses a sophisticated scheme to determine when information about descendant segments is suppressed from the display based on how closely the descendant segments are rendered. When there are many descendant segments in a level, the labels for each segment's thresholds are suppressed, but the segment's split variable continues to be displayed. Where there are even more descendant segments in a level, the label for the segment's split variable is also suppressed. With additional descendant segments, the shape representing the segment in the display is drawn more compactly to permit more segments being rendered side-by-side.

(On page 10, lines 3-9) Please amend as follows:

As described above the condition path, i.e. the path of segments from the top ~~of the strategy~~ of the strategy to the focus segment, is a critical piece of information to the viewer of a strategy in identifying which branch of the strategy is currently displayed and where the branch is located within the strategy. The technique

herein disclosed always displays the set of conditions needed to reach the single segment currently selected as the focus segment.

(On page 11, lines 22-23) Please replace

Design Principle 7: **Provide navigational cues when hopping from one segment to another.**

with

Design Principle 7: Provide navigational cues when hopping from one segment to another.

(On page 12, lines 4-9) Please amend as follows:

An example of this animation is when ~~from~~ hopping from the focus segment to one of the focus segment's descendants. A number of the segments visible on the display initially continue to be displayed after the descendant becomes the new focus segment, but they are located in different positions in the display. The animation shows each of these segments starting in their original position and gradually migrating to their new position.

(On page 12, line 15 through page 13, line 2) Please amend as follows:

Some approaches to strategy visualization can be found in ~~(see, for example, Lamping, et al. 1995)~~ Lamping, J., Rao, R., and Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for viewing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, ACM, the top ~~of the strategy~~ of the strategy is placed dynamically anywhere in the display. The orientation from a node to its descendant nodes can vary its direction. Such approach, while useful from a purely quantitative display point-of-view, is not useful to the strategy consultant, who is used to viewing strategies that have, for example, the top of the strategy

placed in a predictable, consistent location. As such, the herein disclosed approach is configured to adhere to this constraint, *i.e.* the focus segment [[s]] is always at the center, left most portion of the display. The descendant segments of the focus segment always appear to the right of the focus segment.

(On page 15, lines 12-20) Please amend as follows:

Suppose that one wants to navigate to the parent segment of the "MonthsOnBooks" split so one can edit this split. In the invention herein disclosed, one simply clicks the left arrow button to "hop" to it. If one wants to navigate to a ~~sibling~~ sibling, one clicks the up arrow button or the down arrow button. Navigation is simple because one can hop from node to node. In the conventional strategy viewer, one must use the scroll bars to reposition one's view of the canvas. This means that one has to locate (or remember) where on the canvas the node for which one is looking is displayed. For large strategies, this is like finding a needle in a haystack.

(On page 16, lines 1-12) Please amend as follows:

- **Speed**: It was important to implement the invention quickly. Industry studies indicate that with Java programmer productivity is enhanced by at least a factor of five over languages such as C++. This was borne out in the inventor's experience with the herein disclosed invention. The richness of the Java platform, such as automatic double-buffering for animations and automatic anti aliasing for smoother-looking graphics, made it possible to build the software in which the invention is implemented very rapidly. The invention, in its current state, was implemented by a single developer (a complete newcomer to Java) working half time for less than two months. The ~~inventor's~~ inventors are of the opinion that even a deeply experienced C++ developer could not deliver the same level of functionality in so robust a package in such a timeframe.